

**AMENDMENT TO THE TITLE:**

Please amend the title as follows:

Method, System and Mechanism to Multiplex Multiple Application Server Requests Over A Single Database Connection

**AMENDMENT TO THE SPECIFICATION:**

Please replace the paragraph beginning on Page 10, line 9 and continuing to Page 11, line 9, with the following amended version:

-- Figure 6 is a flowchart illustrating the use of the query identifier to avoid section collision for multiple open cursors statements for the same cursor name multiplexed over a single database connection. Figure 7 is a block diagram illustrating the use of query identifiers to avoid section collision. Referring to both Figures 6 and 7, the application server 301 sends a first statement, such as “OPEN C1”, to open a cursor for an application 303 to the database server 302 using a database connection between the application server 301 and the database server 302, via step 601. Upon receiving the first statement, the database server 302 creates a first instance 701 of the cursor, C1, and assigns a first query identifier, QID1, to the first instance 701, via step 602. The database server 302 returns the first query identifier, QID1, to the application server 301, via step 303. Then, the application server 301 sends a second statement, “OPEN C1”, to open the same cursor to the database server 302 using the same database connection, via step 604, before the first instance 701 of the cursor is closed. Upon receiving the second statement, the database server 302 creates a

second instance 702 of the cursor and assigns a second query identifier, QID2, to the second instance 702, via step 605. The database server 302 returns the second query identifier, QID2, to the application server 301, via step 606.

Subsequent statements from the application server 301 for the first instance 701 of the cursor are then sent with the first query identifier, QID1, via step 607.

Subsequent statements from the application server 301 for the second instance 702 of the cursor are then sent with the second query identifier, QID2, via step 608. In combination with the other parameters sent with each statement, the query identifier uniquely identifies ~~identifiers~~ the intended instance of an open cursor. Thus, the query identifiers allow multiple instances of the same cursor to be processed in parallel with each other without section collision. --